

# Đối tượng và Lớp

---

cuu duong than cong . com

Nguyễn Khắc Huy

cuu duong than cong . com

BMCNPM – ĐHKHTN TPHCM  
09/2014



# Nội dung

- Khái niệm đối tượng và lớp.
- Tầm vực.
- Nạp chồng hàm.
- Bài tập.

cuu duong than cong . com

cuu duong than cong . com



# Nhắc lại

- Mục tiêu của việc thiết kế 1 phần mềm:
  - Tính tái sử dụng (reusability): thiết kế các thành phần có thể được sử dụng trong nhiều phần mềm khác nhau
  - Tính mở rộng (extensibility): hỗ trợ các plug-ins.
  - Tính mềm dẻo (flexibility):
    - ✓ Có thể dễ dàng thay đổi khi thêm mới dữ liệu hay tính năng.
    - ✓ Các thay đổi không làm ảnh hưởng nhiều đến toàn bộ hệ thống



# Quá trình thiết kế phần mềm

- ❑ Mục tiêu: xây dựng nên 1 phần mềm
- ❑ Quá trình thiết kế: chia phần mềm và thiết kế theo từng phần, từng component
- ❑ Trừu tượng hóa: rất cần thiết trong thiết kế. Cụ thể: bỏ qua những chi tiết của component, quan tâm các thành phần ở mức trừu tượng.
- ❑ Xác định các component: top-down
- ❑ Tích hợp: bottom-up. Gắn kết các components nhỏ lại với nhau.



# Các cách tiếp cận trong thiết kế

- Thiết kế theo hàm/thủ tục:
  - Tìm ra các hàm/thủ tục để hoàn tất các yêu cầu
  - Kết quả là hệ thống cấu trúc và mối quan hệ giữa các hàm/thủ tục
- Thiết kế theo module
  - Phân tích và tìm ra các module bao gồm thành phần dữ liệu và các hàm/thủ tục liên quan
  - Cách thực hiện dựa vào việc gom nhóm các thành phần tương tự nhau về ý nghĩa, phạm vi... com



# Thiết kế theo hướng đối tượng

- ❑ Trừu tượng hóa dữ liệu và các hàm/thủ tục liên quan
- ❑ Chia hệ thống ra thành các lớp/đối tượng
- ❑ Mỗi lớp/đối tượng có các tính năng và hành động chuyên biệt
- ❑ Các lớp có thể được sử dụng để tạo ra nhiều đối tượng cụ thể

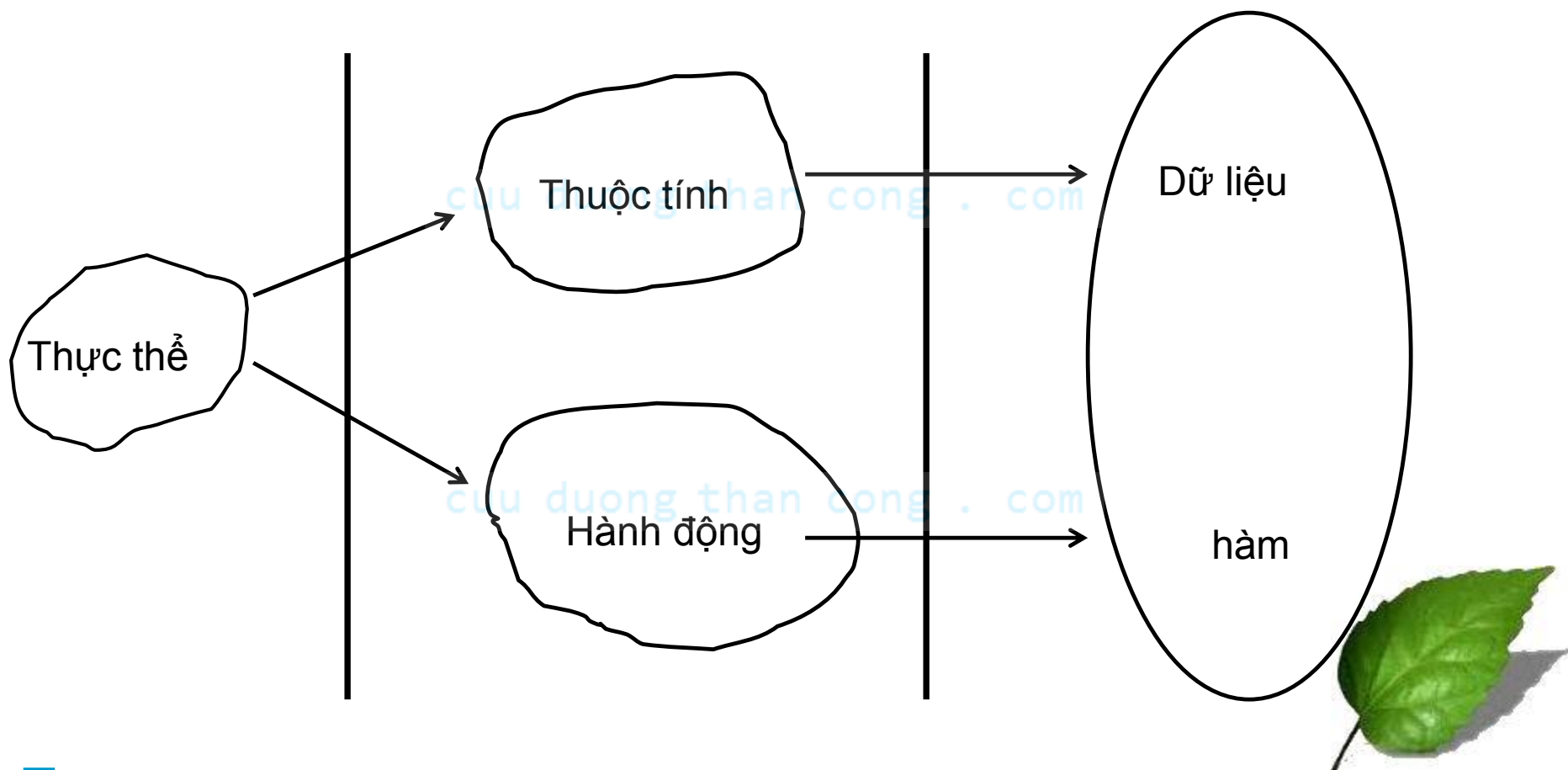


# Trừu tượng hóa

Thế giới thực

Trừu tượng hóa

Phần mềm



# Lập trình hướng đối tượng

Lập trình hướng đối tượng là phương pháp lập trình dựa trên kiến trúc **lớp** (class) và **đối tượng** (object)



# Đối tượng (object) là gì?

- Đối tượng trong thế giới thực: là một thực thể cụ thể mà ta có thể sờ, nhìn thấy hay cảm nhận được.
- Đối tượng phần mềm: dùng để biểu diễn các đối tượng trong thế giới thực.
- Mỗi đối tượng bao gồm 2 thành phần: thuộc tính và hành động.

cuu duong than cong . com



# Khái niệm đối tượng và lớp

- ❑ Chương trình là cỗ máy phức tạp.
- ❑ Cấu thành từ nhiều loại vật liệu.
- ❑ Vật liệu cơ bản: cấu trúc, hàm.
- ❑ Đã đủ tạo ra chương trình tốt?

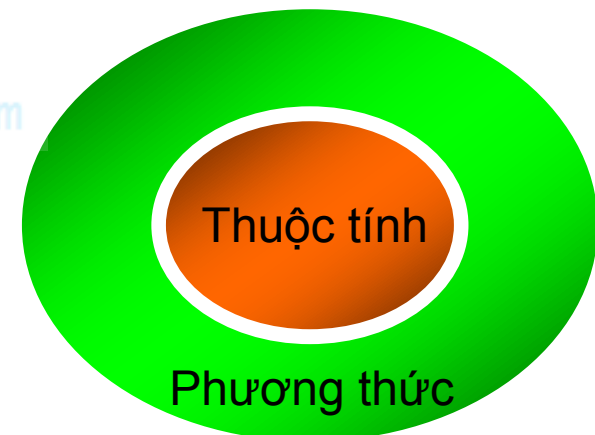


**Vật liệu mới: Đối tượng!!**



# Khái niệm đối tượng và lớp

- Đặc trưng của đối tượng:
  - ✓ Bao gồm cả dữ liệu và xử lý.
  - ✓ Thuộc tính (attribute): dữ liệu của đối tượng.
  - ✓ Phương thức (method): xử lý của đối tượng.
- Cấu trúc đối tượng:
  - ✓ Hộp đen: thuộc tính trong, phương thức ngoài.
  - ✓ Bốn nhóm phương thức:
    - Nhóm tạo hủy.
    - Nhóm truy xuất thông tin.
    - Nhóm xử lý nghiệp vụ.
    - Nhóm toán tử.



# Khái niệm đối tượng và lớp

## □ Vì sao là hướng đối tượng?

✓ Hướng thủ tục (procedural oriented).

– Lấy hành động làm trung tâm.

– Hàm là xương sống.

- Lặt (Rau)

- Ướp (Cá)

- Luộc (Rau)

- Kho (Cá)

- Nấu (Cơm)

✓ Hướng đối tượng (object oriented).

– Lấy dữ liệu làm trung tâm.

– Đối tượng là xương sống.

- Rau.Lặt

- Cá.Ướp

- Rau.Luộc

- Cá.Kho

- Cơm.Nấu

Các bước nấu ăn	
Verb	Object
Lặt	Rau
Ướp	Cá
Nấu	Cơm
Luộc	Rau
Kho	Cá

**Thay đổi  
tư duy  
lập trình!!**

# Khái niệm đối tượng và lớp

## □ Khái niệm lớp:

### Person1:

Name: Peter.  
Age: 25.  
Hair Color: Brown.  
Eye Color: Brown.  
Job: Worker.



### Person2:

Name: Thomas.  
Age: 50.  
Hair Color: White.  
Eye Color: Blue.  
Job: Teacher.



**Tập hợp đối tượng có cùng thuộc tính và phương thức**

### Human:

Name.  
Age.  
Hair Color.  
Eye Color.  
Job.



**Bản mô tả đối tượng  
Kiểu của đối tượng**

# Đối tượng trong LTHĐT

- Tách biệt giữa giao tiếp và cài đặt cụ thể

Làm cái gì?

**interface**

Làm bằng cách nào?

**Implementation**



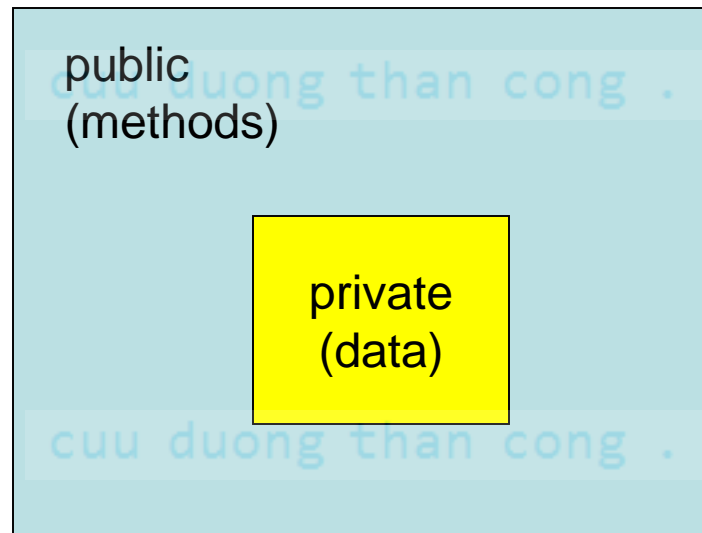
# Đối tượng trong LTHĐT

- Đối tượng là 1 thực thể phần mềm bao bọc các thuộc tính mô tả trạng thái và các phương thức liên quan.
- Kiểu dữ liệu lớp đối tượng: **class**  
**Class** có các tính chất sau:
  - **Đóng gói**: chứa đựng dữ liệu và các hàm/thủ tục liên quan
  - **Che giấu dữ liệu**: các thực thể phần mềm khác không can thiệp trực tiếp vào dữ liệu bên trong được mà phải thông qua các phương thức cho phép



# Một cách thể hiện điển hình

- Che giấu dữ liệu và các “giải thuật” cụ thể ở bên trong lớp (class)



**lớp đối tượng**



# Quy trình thiết kế

- Xác định các đối tượng: xem xét các thực thể trong phạm vi bài toán
- Xác định các hành động tương ứng:
  - Xác định hành động của riêng lớp đối tượng này hay là sự liên hệ của nhiều lớp đối tượng?
  - ...

cuu duong than cong . com



# Một số phương pháp xác định đối tượng

- Theo Abbott & Booch:
  - Tìm các danh từ hay cụm danh từ
  - Không phải tất cả danh từ đều là đối tượng
- Theo Coad & Yourdon:
  - Xác định các cá thể hay các nhóm
- Theo Ross, các đối tượng thường nằm trong các nhóm sau:
  - Người, địa điểm, vật, tổ chức, khái niệm, sự kiện



# Ví dụ

- Xác định các đối tượng
  - Hệ thống âm nhạc lưu trữ các bài hát với tên bài, ca sĩ hát, người sáng tác, xếp hạng, loại bài hát...

cuu duong than cong . com

cuu duong than cong . com



# Lớp đối tượng C++

## □ Sử dụng đối tượng trong C++:

✓ Khai báo lớp: file .h

```
class <Tên lớp>
```

```
{
```

```
    <Khai báo thuộc tính>;
```

```
    <Khai báo phương thức>;
```

```
};
```

✓ Cài đặt phương thức: file .cpp

– Như cài đặt hàm, kèm theo tên lớp.

✓ Tạo đối tượng:

– Như khai báo biến, kiểu dữ liệu là lớp.

✓ Truy xuất thành phần đối tượng:

– Như truy xuất thành phần cấu trúc.



# Ví dụ

## □ Ví dụ:

```
class PhanSo
{
private:
    int m_iTuSo;
    int m_iMauSo;
public:
    PhanSo cong(PhanSo p);
};
```

```
PhanSo PhanSo::cong(PhanSo p)
{
    // Cài đặt cộng phân số...
}
```

```
void main()
{
    PhanSo p1;
    PhanSo *p2 = new PhanSo;
    PhanSo *p3 = new PhanSo[10];

    p3[1] = p1.cong(p3[5]);
    p3[1] = p2->cong(p3[5]);
}
```



# Nội dung

- Khái niệm đối tượng và lớp.
- **Tầm vực.**
- Nạp chồng hàm.
- Bài tập.

cuu duong than cong . com

cuu duong than cong . com



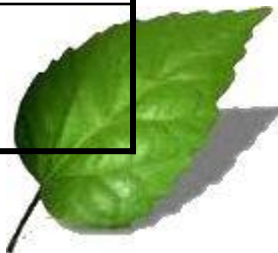
# Phạm vi hoạt động (scope)

## □ Khái niệm tầm vực:

- ✓ Tầm ảnh hưởng, phạm vi hoạt động của thuộc tính, phương thức.
- ✓ Phạm vi càng hẹp, truy xuất càng hạn chế.

## □ Phân loại tầm vực:

Tầm vực	Tầm ảnh hưởng	Phạm vi hoạt động
private	Hẹp	Bên trong lớp.
public	Rộng	Bên trong lẫn bên ngoài lớp.
protected	Vừa	Bên trong lớp và lớp kế thừa.



# Phạm vi hoạt động (scope)

## □ Dr. Guru:

- ✓ Các thành phần trong lớp nên có tầm vực như sau:
  - Thuộc tính có tầm vực private để hạn chế truy xuất.
  - Phương thức có tầm vực public để cung cấp tính năng.

cuu duong than cong . com

```
class PhanSo
{
private:
    int m_iTuSo;
    int m_iMauSo;
public:
    PhanSo cong(PhanSo p);
};
```



# Phạm vi hoạt động (scope)

## □ Toán tử :: :

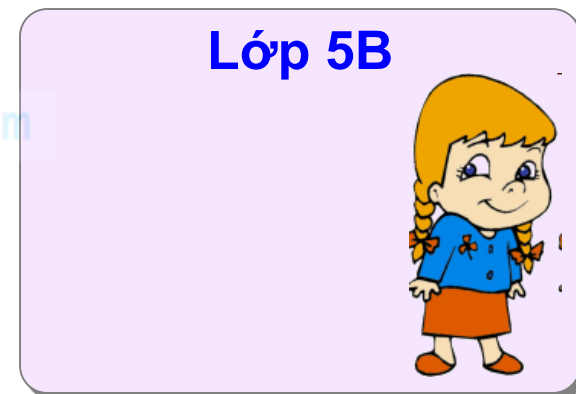
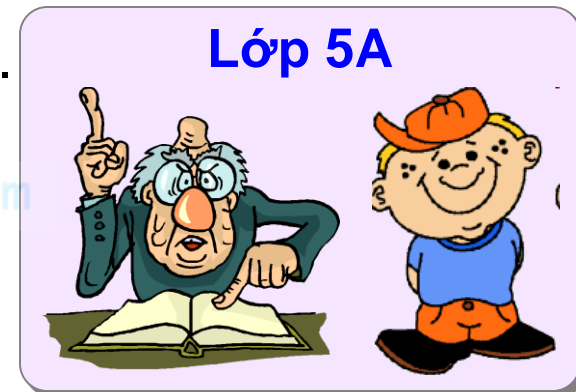
✓ Lớp 5A và 5B đều có sinh viên Lê Phương.

✓ **Làm sao để phân biệt?**

- Trường hợp 1: đứng trong lớp 10A1.
- Trường hợp 2: đứng ngoài 2 lớp.

✓ **Toán tử :: dùng kèm tên lớp.**

- <Tên lớp>::<Tên thành phần>



# Toán tử phạm vi ::

- Dùng để chỉ dẫn thành phần thuộc tính hay phương thức thuộc một lớp đối tượng cụ thể nào đó

Ví dụ:

```
Ngay::layNgay()  
Ngay::layThang()
```



# Toán tử phạm vi ::

□ Ví dụ:

```
class PhanSo
```

```
{
```

```
private:
```

```
    int m_iTuSo;
```

```
    int m_iMauSo;
```

```
public:
```

```
    PhanSo cong(PhanSo p)
```

```
{
```

```
    // Cài đặt bên trong lớp.
```

```
}
```

```
};
```

```
PhanSo PhanSo::cong(PhanSo p)
```

```
{
```

```
    // Cài đặt bên ngoài lớp.
```

```
}
```



# Toán tử phạm vi ::

## □ Con trỏ this:

- ✓ Mỗi lớp đều có con trỏ this.
- ✓ Đại diện cho đối tượng đang gọi phương thức.
- ✓ Hữu dụng trong một số trường hợp.

```
class PhanSo
{
private:
    int    m_iTuSo;
    int    m_iMauSo;
public:
    void ganTuSo(int iTuSo) { this->m_iTuSo = iTuSo; }
};
```

```
void main()
{
    PhanSo p1;
    p1.ganTuSo(3);

    PhanSo p2;
    p2.ganTuSo(5);
}
```



# Nội dung

- Khái niệm đối tượng và lớp.
- Tầm vực.
- Nạp chồng hàm.
- Bài tập.

cuu duong than cong . com

cuu duong than cong . com



# Nạp chồng hàm

## □ Chữ ký hàm (Function Signature):

- ✓ Phân biệt giữa các hàm.
- ✓ Đặt trưng của một hàm:
  - Tên hàm.
  - Danh sách tham số.

```
double sapXep(int aMang[], int iKichThuoc);  
PhanSo PhanSo::cong(PhanSo p);
```

✓ **Giá trị trả về không thuộc chữ ký hàm!!**

**Chương trình có thể có  
nhiều hàm cùng tên!!**



# Nạp chồng hàm

□ Những trường hợp nạp chồng hàm nào không hợp lệ?

1. `int tinhToan(int a, int b);`
2. `int tinhToan(int x, int y);`
3. `int tinhToan(int a, float b);`
4. `float tinhToan(int u, int v);`
5. `int tinhToan(int a, long b);`



# Tóm tắt

## □ Khái niệm đối tượng và lớp:

- ✓ Đối tượng: một thành phần tạo nên chương trình.
- ✓ Đối tượng bao gồm thuộc tính và phương thức.
- ✓ Lớp: tập hợp đối tượng cùng thuộc tính phương thức.

[cuu duong than cong . com](http://cuuduongthancong.com)

## □ Tầm vực:

- ✓ Phạm vi hoạt động của thuộc tính và phương thức.
- ✓ Có 3 mức: public, private, protected.

## □ Nạp chồng hàm:

- ✓ Nhiều hàm cùng tên, khác tham số.



# Tách biệt khai báo và định nghĩa

```
class Ngay
{
    private:
        ...
    public:
        int layNgay()
        {
            return iNgay;
        }
};
```

```
// trong file header .H
class Ngay
{
    private:
        ...
    public:
        int layNgay();
};

//trong file .CPP
int Ngay::layNgay()
{
    return iNgay;
}
```

# Tách thành phần khai báo và cài đặt

- File header (interface): [Ngay.h](#)  
Chứa các thông tin về khai báo của lớp Ngày
- File xử lý (implementation): [Ngay.cpp](#)  
Chứa các cài đặt về xử lý của lớp Ngày

[cuu duong than cong . com](#)



# Nội dung

- Khái niệm đối tượng và lớp.
- Tầm vực.
- Nạp chồng hàm.
- Bài tập.

cuu duong than cong . com

cuu duong than cong . com

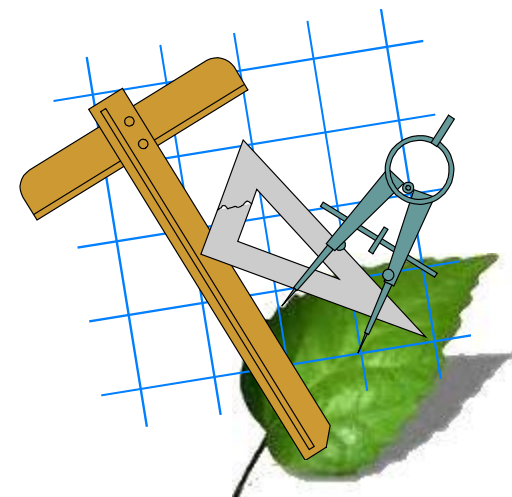


# Bài tập

## □ Bài tập 2.1:

Xây dựng lớp **phân số** cho phép thực hiện các thao tác:

- ✓ Nhập, xuất.
- ✓ Lấy tử số, mẫu số.
- ✓ Gán giá trị cho tử số, mẫu số.
- ✓ Nghịch đảo, rút gọn.
- ✓ Cộng, trừ, nhân, chia với phân số khác.

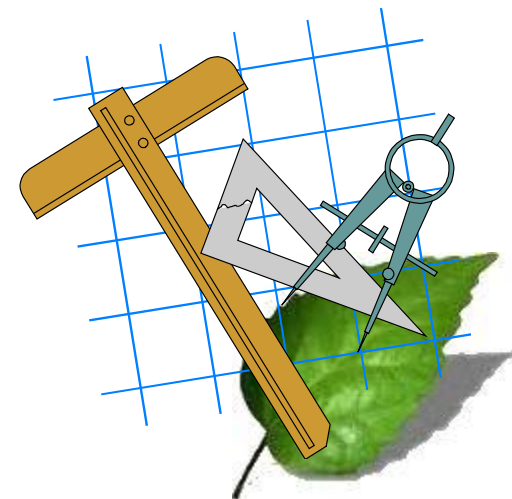


# Bài tập

## □ Bài tập 2.2:

Xây dựng lớp **số phức** cho phép thực hiện các thao tác:

- ✓ Nhập, xuất.
- ✓ Lấy phần thực, phần ảo.
- ✓ Gán giá trị cho phần thực, phần ảo.
- ✓ Tính module.
- ✓ Cộng, trừ, nhân, chia với số phức khác.

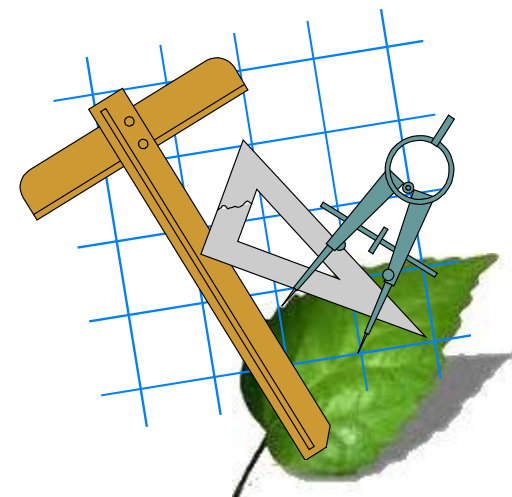


# Bài tập

## □ Bài tập 2.3:

Xây dựng lớp **đơn thức** cho phép thực hiện các thao tác:

- ✓ Nhập, xuất.
- ✓ Lấy hệ số, số mũ.
- ✓ Gán giá trị cho hệ số, số mũ.
- ✓ Tính giá trị, đạo hàm, nguyên hàm.
- ✓ Cộng, trừ, nhân, chia với đơn thức khác.



# Bài tập

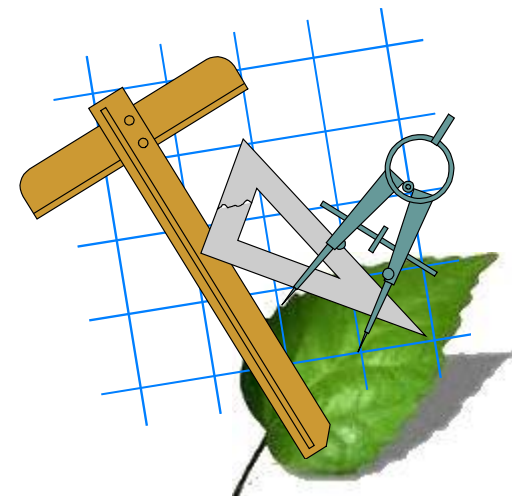
## □ Bài tập 2.4:

Thông tin một học sinh bao gồm:

- ✓ Họ tên.
- ✓ Điểm văn, toán.

Xây dựng lớp **học sinh** cho phép thực hiện các thao tác:

- ✓ Nhập, xuất.
- ✓ Lấy họ tên, điểm văn, toán.
- ✓ Gán giá trị cho họ tên, điểm văn, điểm toán.
- ✓ Tính điểm trung bình.
- ✓ Xếp loại theo tiêu chí
  - Giỏi ( $\geq 8.0$ ), Khá ( $\geq 7.0$ ).
  - Trung bình ( $\geq 5.0$ ), Yếu ( $< 5$ ).

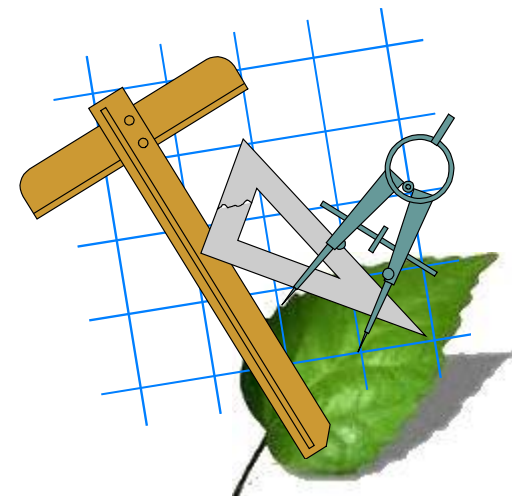


# Bài tập

## □ Bài tập 2.5:

Xây dựng lớp **mảng** cho phép thực hiện các thao tác sau:

- ✓ Nhập, xuất mảng.
- ✓ Lấy kích thước mảng.
- ✓ Lấy phần tử tại vị trí nào đó.
- ✓ Gán giá trị cho phần tử tại vị trí nào đó.
- ✓ Tìm phần tử nào đó trong mảng.
- ✓ Sắp xếp tăng, giảm.



# Lời cảm ơn

- Nội dung được xây dựng dựa trên slide trình bày của Thầy Đinh Bá Tiến, Thầy Nguyễn Minh Huy.

cuu duong than cong . com

cuu duong than cong . com

